Answers to Review Questions
Chapter 6

1. Because they are created in memory when the function begins execution, and are destroyed when the function ends.

2. An argument is a value passed to a function. A parameter variable is a variable local to the function which receives the argument. That is to say, the argument's value is copied into the parameter variable.

3. Inside the parentheses of a function header.

4. If the argument is passed by value, nothing needs to be done. The function cannot access the argument. If the argument is passed by reference, the parameter should be defined with the `const` key word.

5. Yes. The first argument is passed into the parameter variable that appears first inside the function header's parentheses. Likewise, the second argument is passed into the second parameter, and so on.

6. With the `return` statement. For example, the following statement returns the value in the variable `x`.

   ```
   return x;
   ```

7. It makes the program easier to manage. Imagine a book that has a thousand pages, but isn’t divided into chapters or sections. Trying to find a single topic in the book would be very difficult. Real-world programs can easily have thousands of lines of code, and unless they are modularized, they can be very difficult to modify and maintain.

8. It would enable a function to retain a value between function calls. For example, it can keep track of the number of times the function has been called.

9. A function such as the following could be written to get user input. The input is stored in the variables that are passed as arguments.

   ```
   void getValues(int &x, int &y)
   {
        cout << "Enter a number: ";
        cin >> x;
        cout << "Enter another number: ";
        cin >> y;
   }
   ```

10. header

11. `void`
12. definition, prototype
13. arguments
14. parameters
15. value
16. prototype
17. local
18. global
19. global
20. 0
21. local
22. `static`
23. `return`
24. default
25. last
26. constant
27. reference
28. &
29. reference
30. `exit`
31. parameter lists
32. `showValue(x);`

33.

```
double half(double num)
{
     return num / 2;
}
```

34. `result = cube(4);`

35.

```
void timesTen(int num)
{
     cout << (num * 10) << endl;
}
```

36. `display(age, income, initial);`

37.

```
void getNumber(int &num)
{
      cout << "Enter a number in the range 1 – 100 : ";
      cin >> num;
      while (num < 1 || num > 100)
      {
            cout << "That number is out of range.\n";
            cout << "Enter a number in the range 1 – 100 : ";
            cin >> num;
      }
}
```

38. true
39. false
40. true
41. true
42. false
43. true
44. true
45. false
46. true
47. true
48. true
49. true
50. false
51. true
52. true
53. false
54. false
55. true
56. The data type of `value2` and `value3` must be defined.
    The function is declared `void` but returns a value.
57. The assignment statement should read:
    ```
        average = (value1 + value2 + value3) / 3.0;
    ```
    The function is defined as a `float` but returns no value.
58. `width` should have a default argument value.
    The function is defined `void` but returns a value.
59. The parameter should be defined as:
    ```
        int &value
    ```
    The `cin` statement should read:
    ```
        cin >> value;
    ```
60. The functions must have different parameter lists.